Maximum Entropy Framework: Inverse RL, Soft Optimality, and More

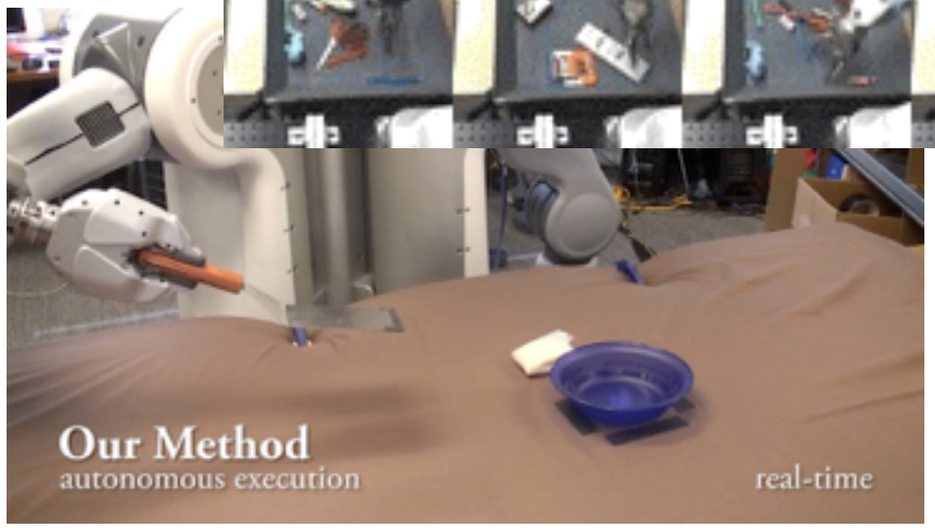
Chelsea Finn and Sergey Levine
UC Berkeley

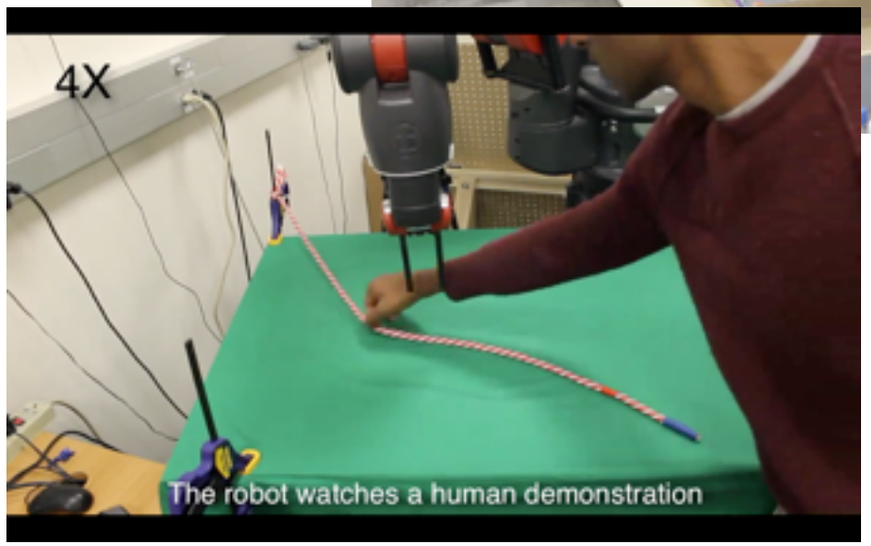
5/20/2017

Introductions

Chelsea Finn
PhD student

Sergey Levine
assistant professor





Outline

1. A World without Rewards
2. A Probabilistic Model of Behavior
3. Application: Inverse RL
4. GANs and Energy-Based Models
5. Application: Soft-Q Learning

Outline

1. A World without Rewards

2. A Probabilistic Model of Behavior

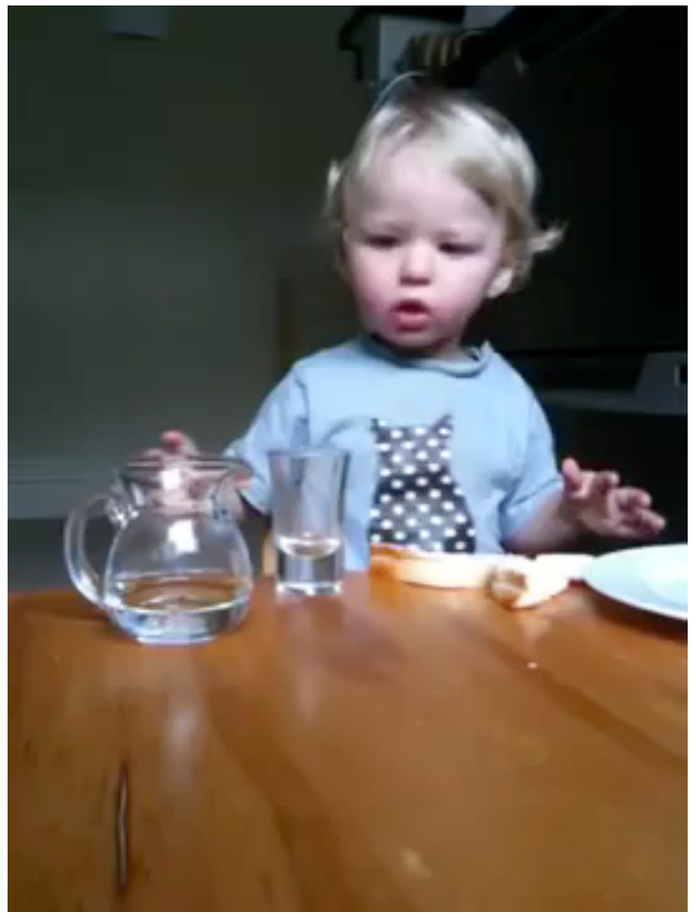
3. Application: Inverse RL

4. GANs and Energy-Based Models

5. Application: Soft-Q Learning

reward

Mnih et al. ’15

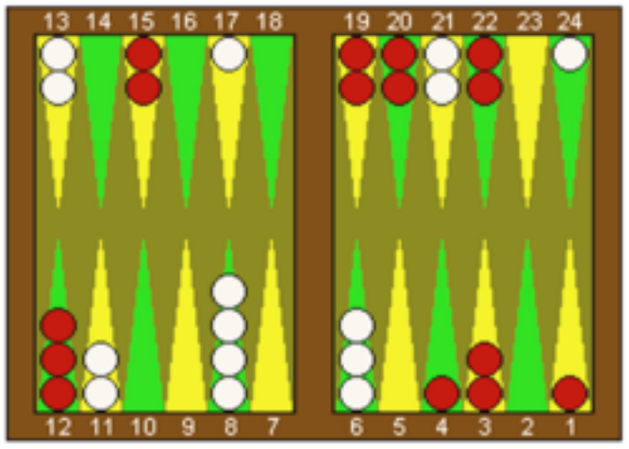


reinforcement learning agent

what is the reward?

In the real world, humans don’t get a score.

video from Montessori New Zealand



Tesauro ’95

Kohl & Stone, ’04

Mnih et al. ’15

AlphaGo

Silver et al. ‘16

reward function is essential for RL

real-world domains: reward/cost often difficult to specify

- robotic manipulation
- autonomous driving
- dialog systems
- virtual assistants
- and more…

One approach: Mimic actions of human expert

behavioral cloning + simple, sometimes works well

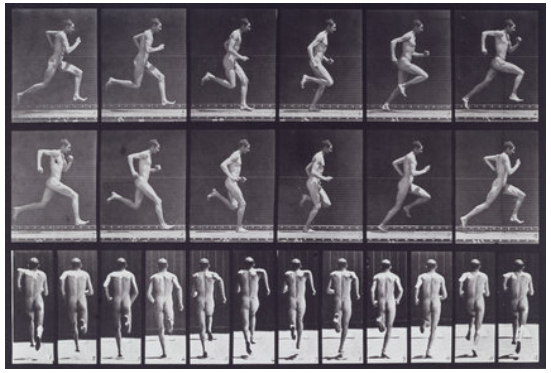
- but no reasoning about outcomes or dynamics
- the expert might have different degrees of freedom
- the expert might not be always optimal

Can we reason about human decision-making?

Outline

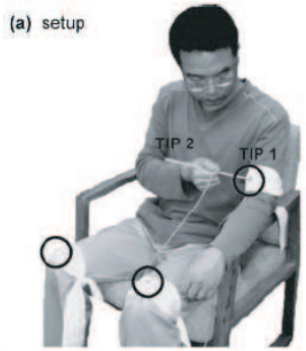
1. A World without Rewards
2. **A Probabilistic Model of Behavior**
3. Application: Inverse RL
4. GANs and Energy-Based Models
5. Application: Soft-Q Learning

Optimal Control as a Model of Human Behavior



Muybridge (c. 1870)

Mombaur et al. '09



Li & Todorov '06

Ziebart '08

$$\mathbf{a}_1, \ldots, \mathbf{a}_T = \arg \max_{\mathbf{a}_1, \ldots, \mathbf{a}_T} \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)$$

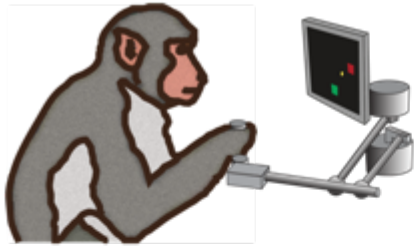
$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$$

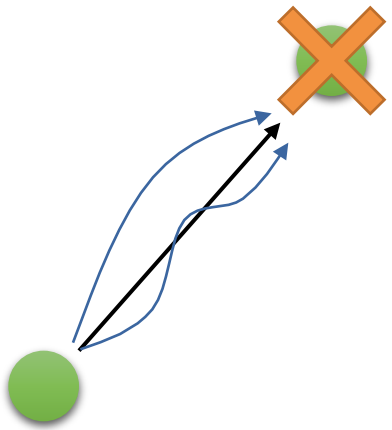
optimize this to explain the data

$$\pi = \arg \max_{\pi} E_{\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t), \mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)} [r(\mathbf{s}_t, \mathbf{a}_t)]$$

$$\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$$

What if the data is **not** optimal?





some mistakes matter more than others!

behavior is **stochastic**

but good behavior is still the most likely

A probabilistic graphical model of decision making

$$\mathbf{a}_1, \ldots, \mathbf{a}_T = \arg\max_{\mathbf{a}_1,\ldots,\mathbf{a}_T} \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)$$

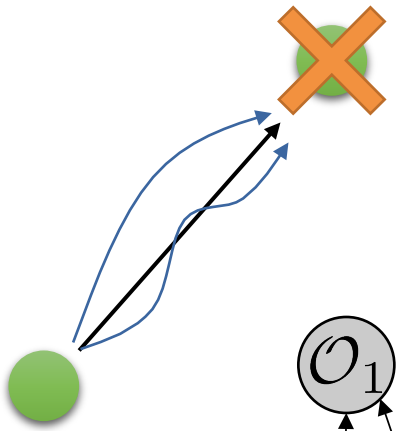
$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$$

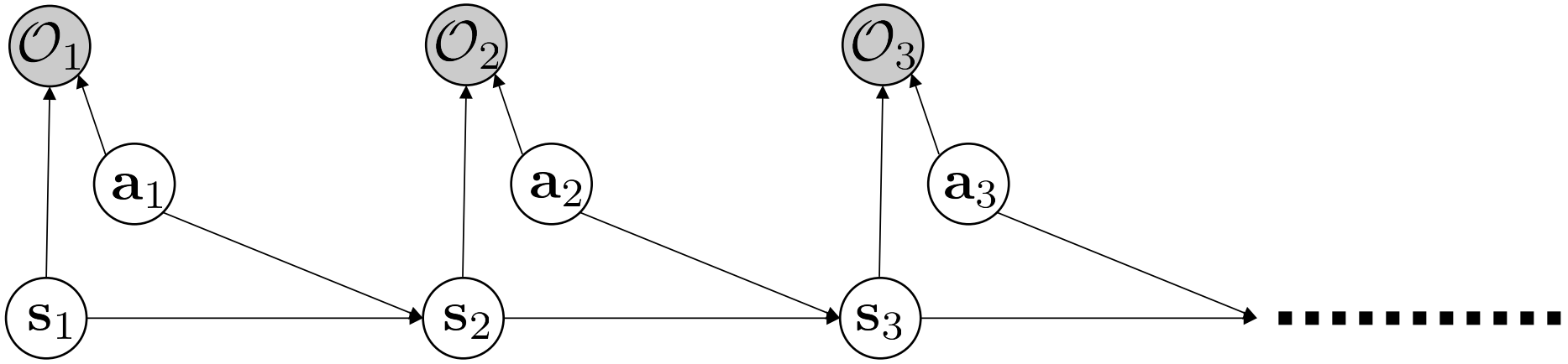
$p(\underbrace{\mathbf{s}_{1:T}, \mathbf{a}_{1:T}}_{\tau}) = ??$ no assumption of optimal behavior!

$p(\tau|\mathcal{O}_{1:T})$ $\quad p(\mathcal{O}_t|\mathbf{s}_t, \mathbf{a}_t) \propto \exp(r(\mathbf{s}_t, \mathbf{a}_t))$

$$p(\tau|\mathcal{O}_{1:T}) = \frac{p(\tau, \mathcal{O}_{1:T})}{p(\mathcal{O}_{1:T})}$$

$$\propto p(\tau) \prod_t \exp(r(\mathbf{s}_t, \mathbf{a}_t)) = p(\tau) \exp\left(\sum_t r(\mathbf{s}_t, \mathbf{a}_t)\right)$$

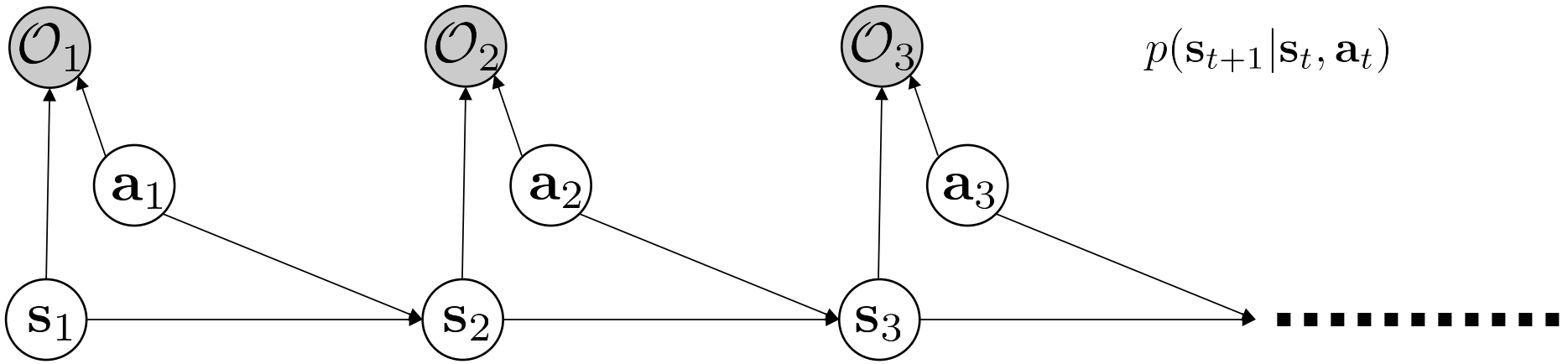




Inference = planning

$$p(\mathcal{O}_t) \propto \exp(r(\mathbf{s}_t, \mathbf{a}_t))$$

$$p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$



how to do inference?

1. compute backward messages $\beta_t(\mathbf{s}_t, \mathbf{a}_t) = p(\mathcal{O}_{t:T}|\mathbf{s}_t, \mathbf{a}_t)$
2. compute policy $p(\mathbf{a}_t|\mathbf{s}_t, \mathcal{O}_{1:T})$
3. compute forward messages $\alpha_t(\mathbf{s}_t) = p(\mathbf{s}_t|\mathcal{O}_{1:t-1})$

A closer look at the backward pass

for $t = T - 1$ to 1:

$$\beta_t(\mathbf{s}_t, \mathbf{a}_t) = p(\mathcal{O}_t|\mathbf{s}_t, \mathbf{a}_t) E_{\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)}[\beta_{t+1}(\mathbf{s}_{t+1})]$$

$$\beta_t(\mathbf{s}_t) = E_{\mathbf{a}_t \sim p(\mathbf{a}_t|\mathbf{s}_t)}[\beta_t(\mathbf{s}_t, \mathbf{a}_t)]$$

"optimistic" transition
(not a good idea!)

let $V_t(\mathbf{s}_t) = \log \beta_t(\mathbf{s}_t)$

let $Q_t(\mathbf{s}_t, \mathbf{a}_t) = \log \beta_t(\mathbf{s}_t, \mathbf{a}_t)$

$$V_t(\mathbf{s}_t) = \log \int \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t)) \mathbf{a}_t$$

$$Q_t(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \log E[\exp(V_{t+1}(\mathbf{s}_{t+1}))]$$

deterministic transition: $Q_t(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + V_{t+1}(\mathbf{s}_{t+1})$

a better stochastic model: $Q_t(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + E[V_{t+1}(\mathbf{s}_{t+1})]$

$V_t(\mathbf{s}_t) \rightarrow \max_{\mathbf{a}_t} Q_t(\mathbf{s}_t, \mathbf{a}_t)$ as $Q_t(\mathbf{s}_t, \mathbf{a}_t)$ gets bigger!

Ziebart et al. '10 "Modeling Interaction via the Principle of Maximum Causal Entropy"

Stochastic optimal control (MaxCausalEnt) summary

for $t = T-1$ to 1:

$$Q_t(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + E[V_{t+1}(\mathbf{s}_{t+1})]$$

$$V_t(\mathbf{s}_t) = \log \int \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t))\mathbf{a}_t$$

variants:

discounted SOC: $Q_t(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma E[V_{t+1}(\mathbf{s}_{t+1})]$

explicit temperature: $V_t(\mathbf{s}_t) = \alpha \log \int \exp\left(\frac{1}{\alpha} Q_t(\mathbf{s}_t, \mathbf{a}_t)\right) d\mathbf{a}_t$

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \frac{\beta_t(\mathbf{s}_t, \mathbf{a}_t)}{\beta_t(\mathbf{s}_t)}$$

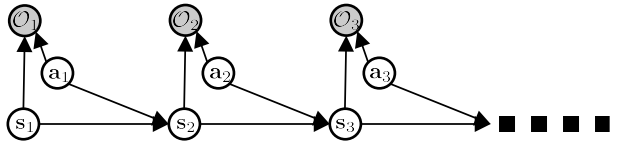
$$V_t(\mathbf{s}_t) = \log \beta_t(\mathbf{s}_t)$$

$$Q_t(\mathbf{s}_t, \mathbf{a}_t) = \log \beta_t(\mathbf{s}_t, \mathbf{a}_t)$$

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t) - V_t(\mathbf{s}_t)) = \exp(A_t(\mathbf{s}_t, \mathbf{a}_t))$$

summary:

1. Probabilistic graphical model for optimal control



2. Control = inference (similar to HMM, EKF, etc.)

3. Very similar to dynamic programming, value iteration, etc. (but “soft”)

Outline

1. A World without Rewards
2. A Probabilistic Model of Behavior
3. **Application: Inverse RL**
4. GANs and Energy-Based Models
5. Application: Soft-Q Learning

Under reward, we can model how human can sub-optimally maximize reward.

How can this help us with learning?

Inverse Optimal Control / Inverse Reinforcement Learning:

infer cost/reward function from demonstrations

(IOC/IRL) (Kalman '64, Ng & Russell '00)

given:
- state & action space
- roll-outs from π^*
- dynamics model [sometimes]

goal:
- recover reward function
- then use reward to get policy

Challenges

underdefined problem

difficult to evaluate a learned reward

demonstrations may not be precisely optimal

Early IRL Approaches

- deterministic MDP
- alternative between solving MDP & updating reward
- heuristics for handling sub-optimality

Ng & Russell '00: expert actions should have higher value than other actions, larger gap is better

Abbeel & Ng '04: expert policy w.r.t. cost should match feature counts of expert trajectories

Ratliff et al. '06: max margin formulation between value of expert actions and other actions

How to handle ambiguity and suboptimality?

Maximum Entropy Inverse RL

(Ziebart et al. '08)

handle ambiguity using probabilistic model of behavior

Notation:

$$\tau = \{s_1, a_1, ..., s_t, a_t, ..., s_T\}$$

r_θ reward with parameters θ [linear case $r_\theta(\tau) = \theta^T \mathbf{f}_\tau = \sum_{s \in \tau} \theta^T \mathbf{f}_s$]

$\mathcal{D}$: dataset of demonstrations $M = |\mathcal{D}|$

Whiteboard

Maximum Entropy Inverse RL

(Ziebart et al. '08)

0. Initialize θ, gather demonstrations $\mathcal{D}$
1. Solve for optimal policy $\pi(a|s)$ w.r.t. r_θ with value iteration
2. Solve for state visitation frequencies $p(s|\theta)$
3. Compute gradient $\nabla_\theta \mathcal{L} = \frac{1}{M} \sum_{\tau_d \in \mathcal{D}} \mathbf{f}_{\tau_d} + \sum_s p(s|\theta) \mathbf{f}_s$
4. Update θ with one gradient step using $\nabla_\theta \mathcal{L}$

What about unknown dynamics?

Whiteboard

Case Study: Guided Cost Learning

Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization

Chelsea Finn CBFINN@EECS.BERKELEY.EDU
Sergey Levine SVLEVINE@EECS.BERKELEY.EDU
Pieter Abbeel PABBEEL@EECS.BERKELEY.EDU

ICML 2016

Goals:

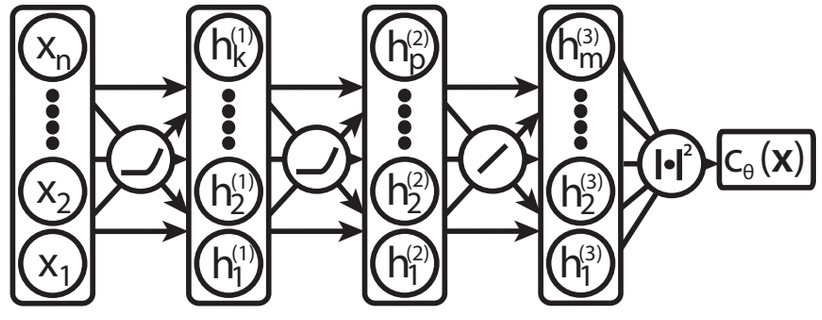
- remove need to solve MDP in the inner loop
- be able to handle unknown dynamics
- handle continuous state & actions spaces

guided cost learning algorithm

initial
policy π_0

human
demonstrations

generate policy
samples from π



Update reward using
samples & demos

update π w.r.t. reward

policy π

reward r

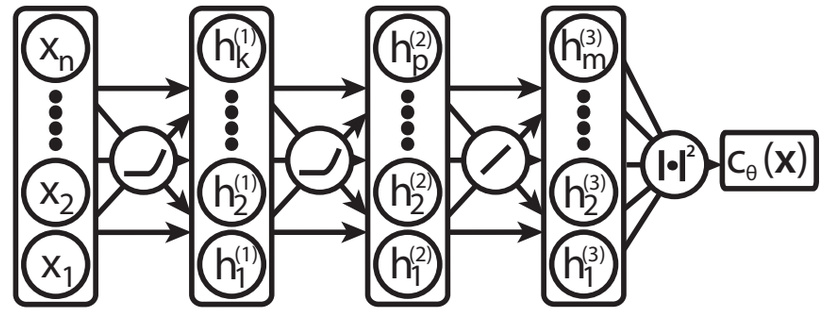
guided cost learning algorithm

initial policy π_0

human demonstrations

generate policy samples from π

generator



Update reward using samples & demos

discriminator

policy π

update π w.r.t. reward (partially optimize)

reward r

update reward in inner loop of policy optimization

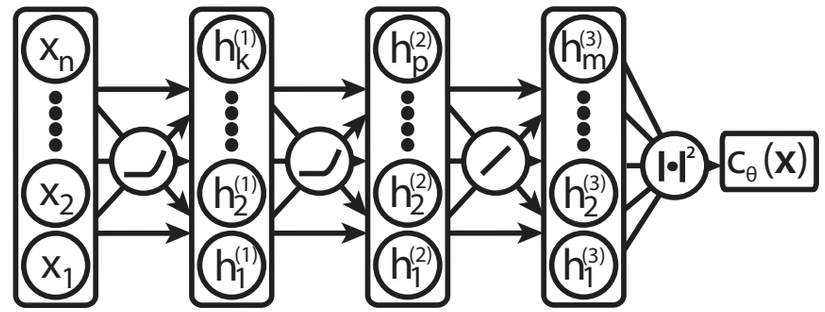
guided cost learning algorithm

initial policy π_0

human demonstrations

generate policy samples from π

generator



Update reward using samples & demos

discriminator

policy π

update π w.r.t. reward (partially optimize)

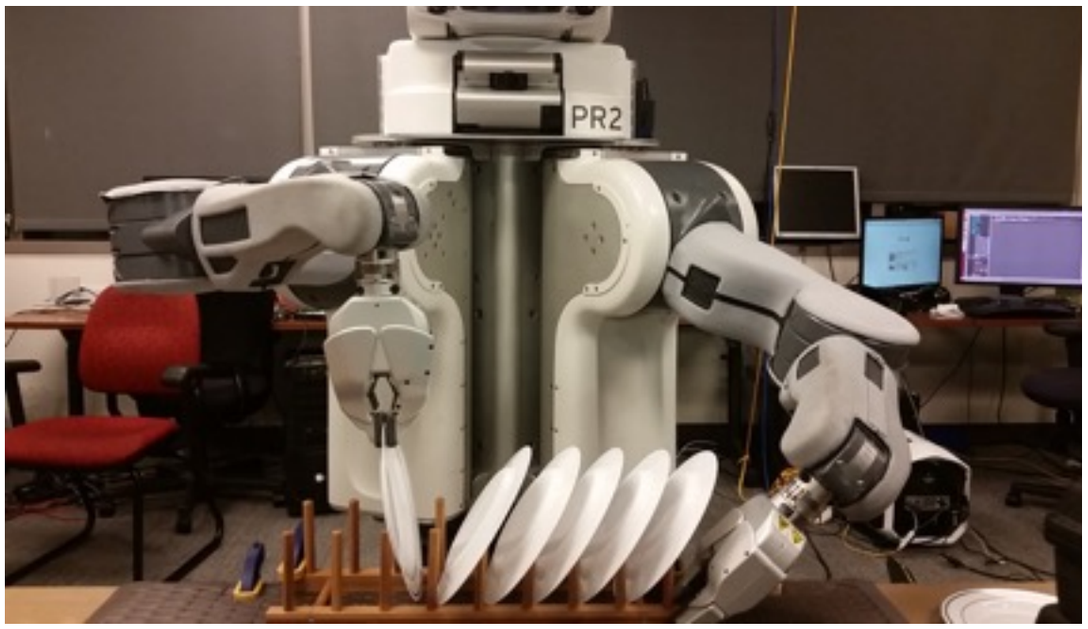
reward r

Ho et al., ICML '16, NIPS '16

GCL Experiments

Real-world Tasks

dish placement



state includes goal plate pose

pouring almonds

state includes unsupervised visual features [Finn et al. ’16]

action: joint torques

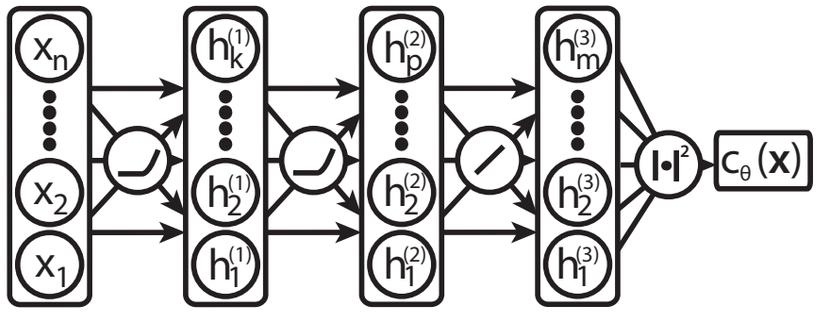
Comparisons

Path Integral IRL (Kalakrishnan et al. '13)

initial distribution q_0

generate policy samples from q

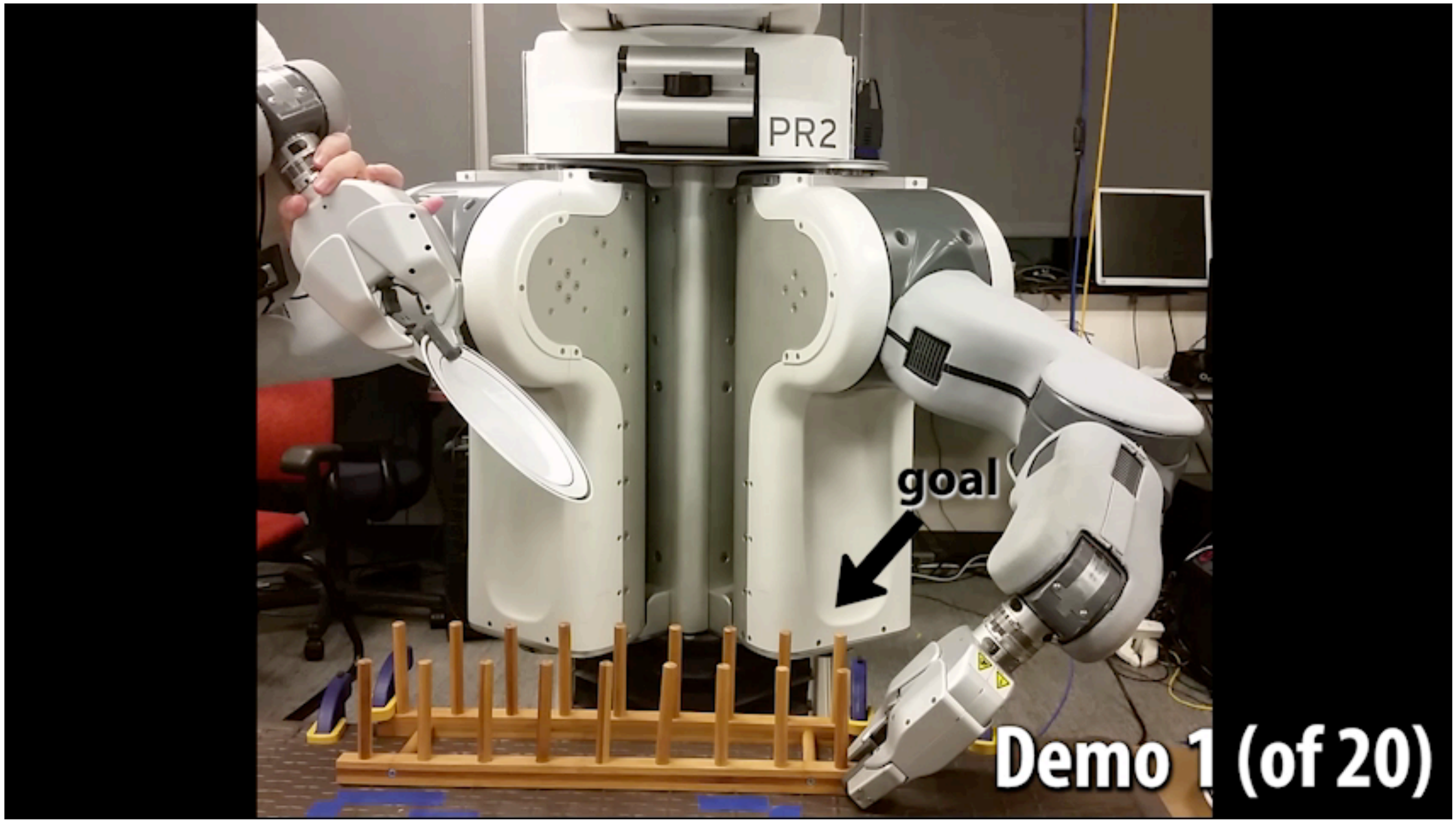
Relative Entropy IRL (Boularias et al. '11)



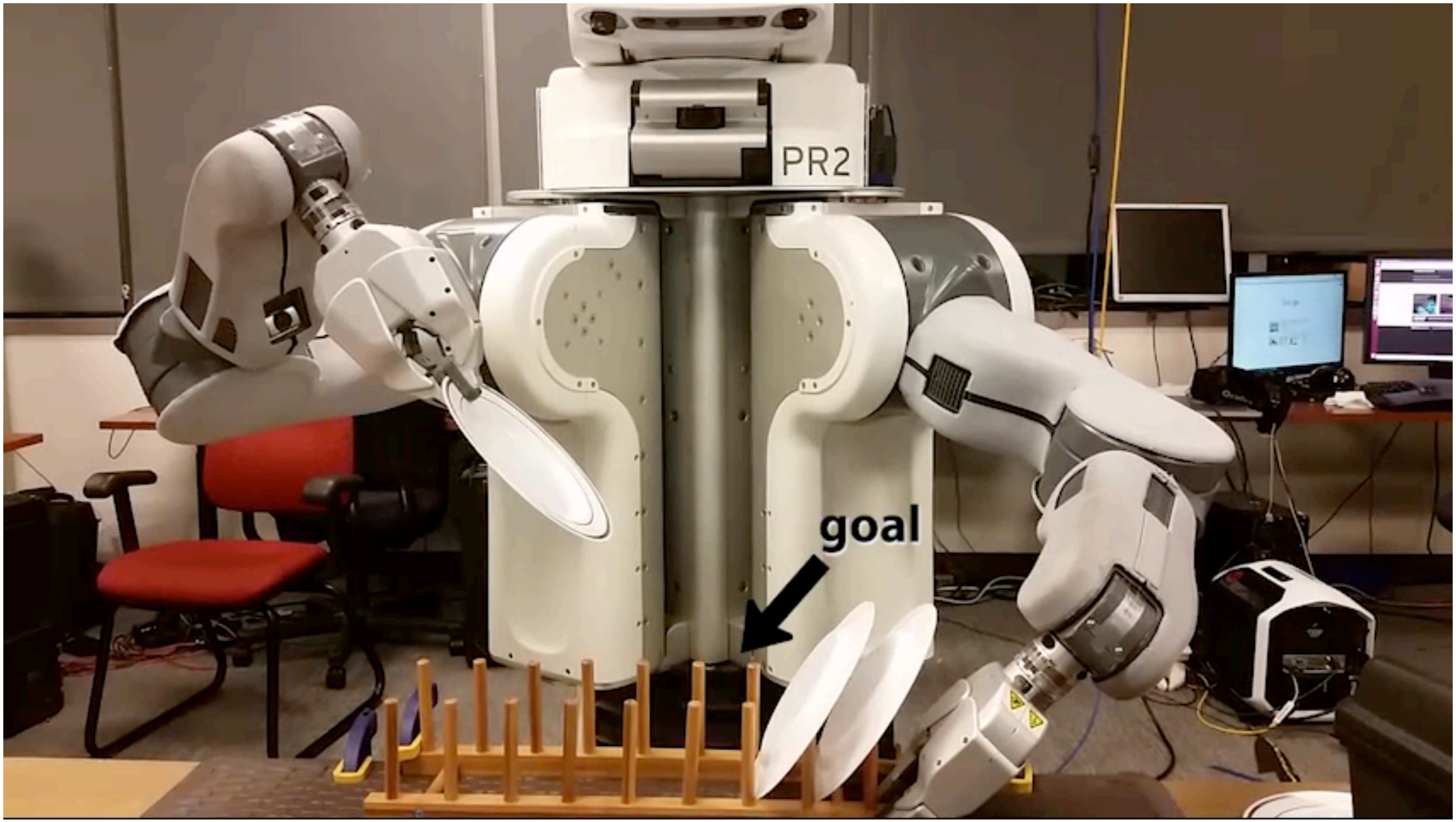
Update reward using samples & demos

reward r

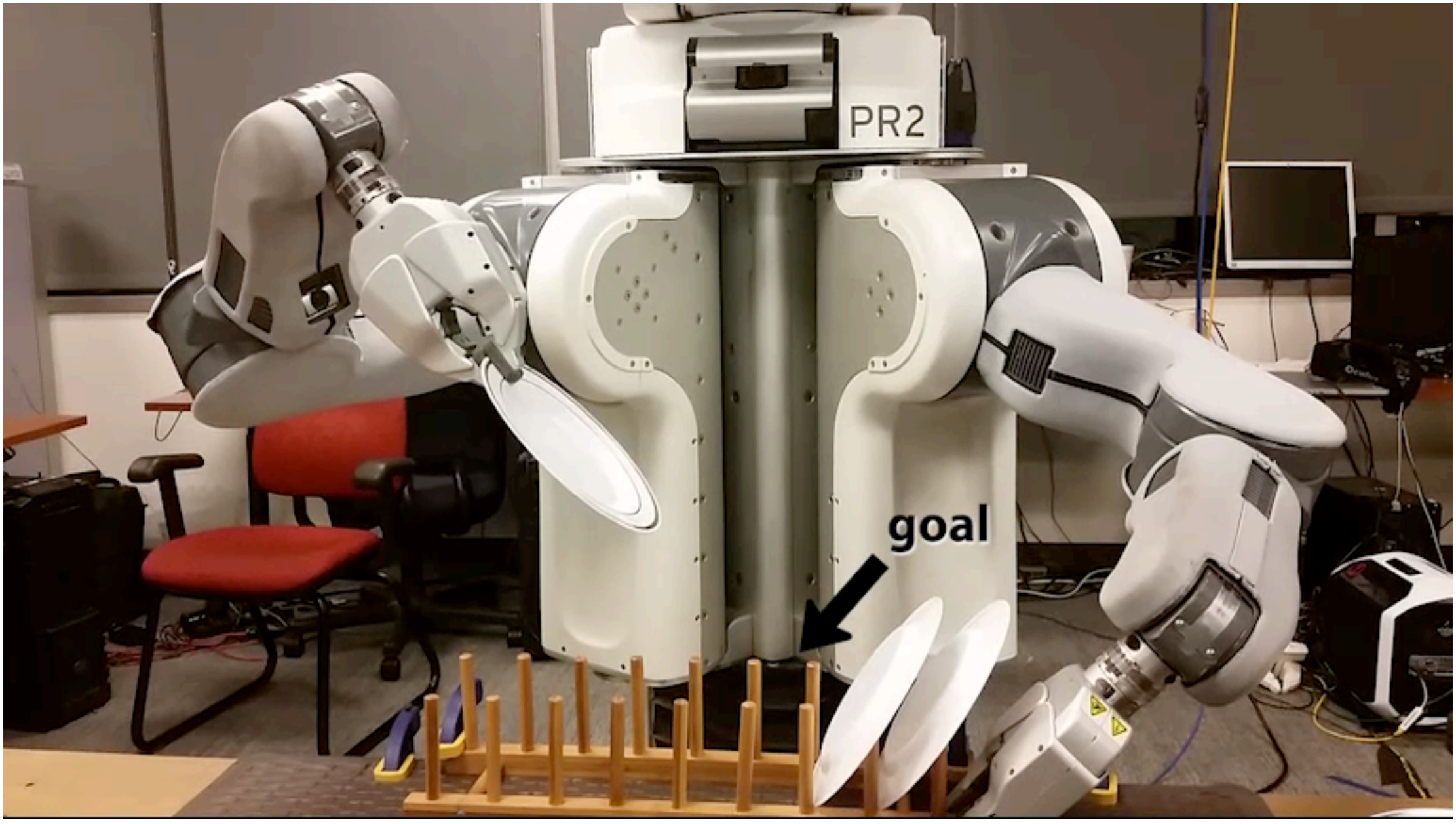
Dish placement, demos



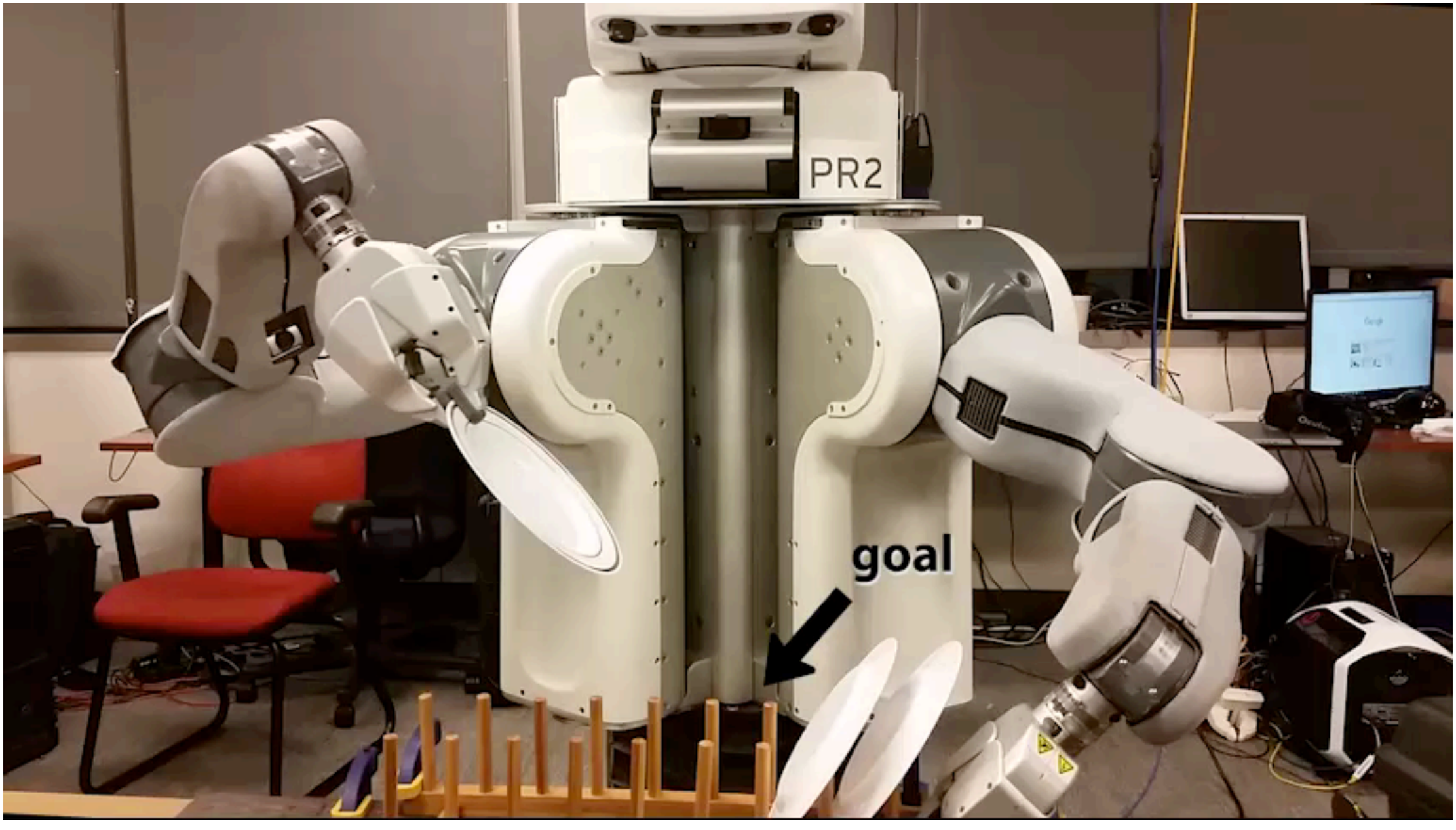
Dish placement, standard cost



Dish placement, RelEnt IRL

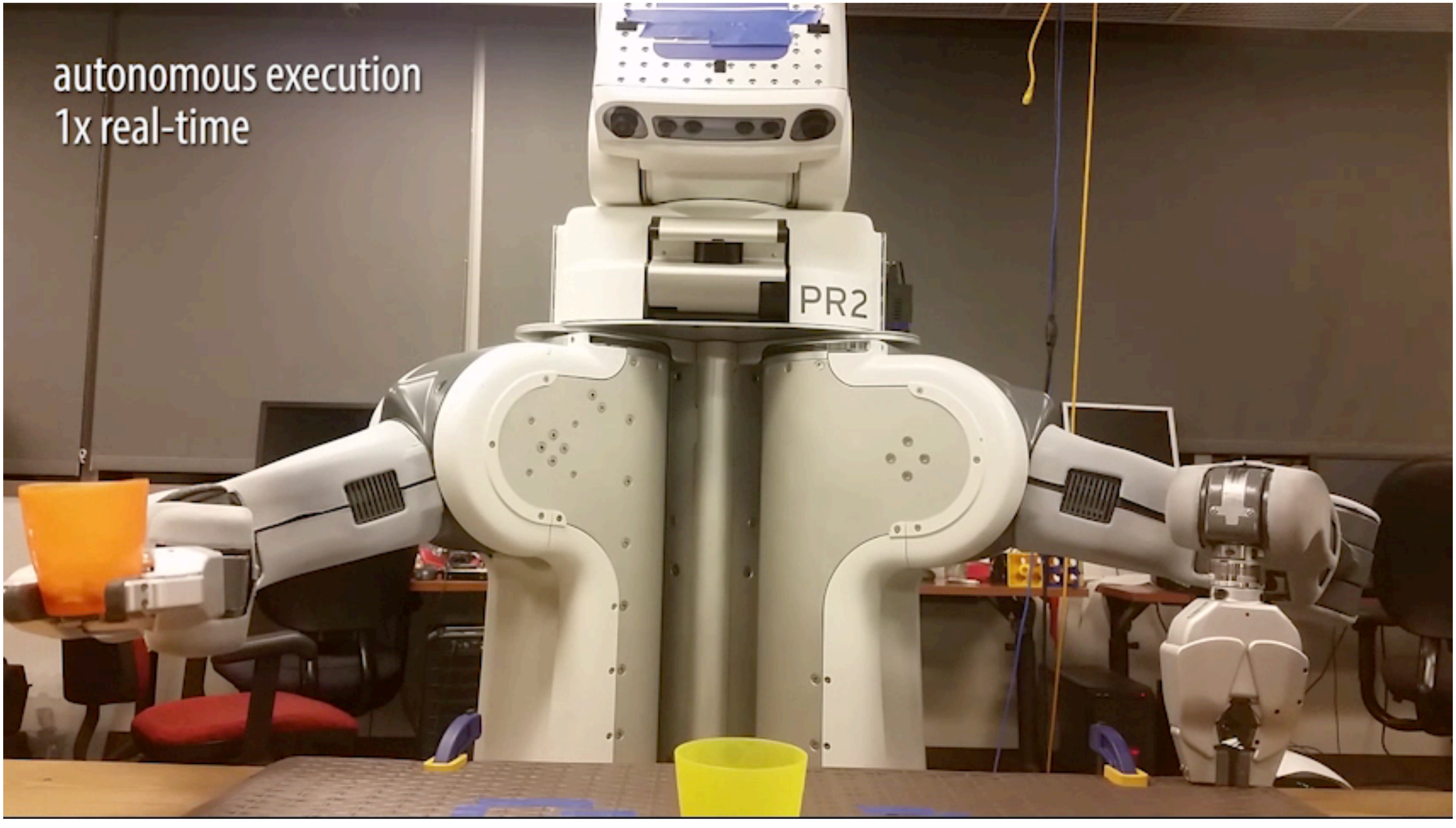


Dish placement, GCL policy

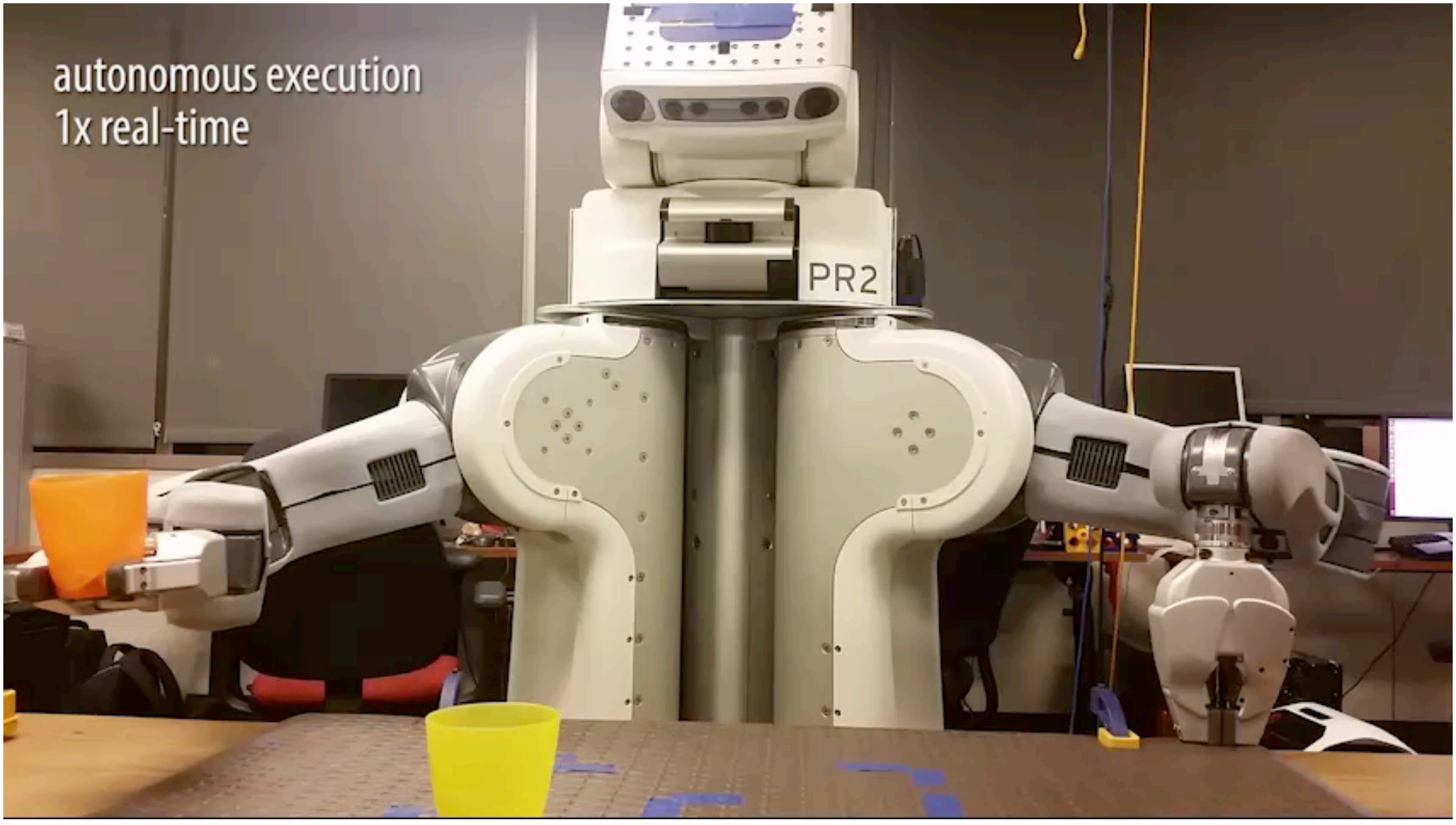


Pouring, demos

Pouring, RelEnt IRL



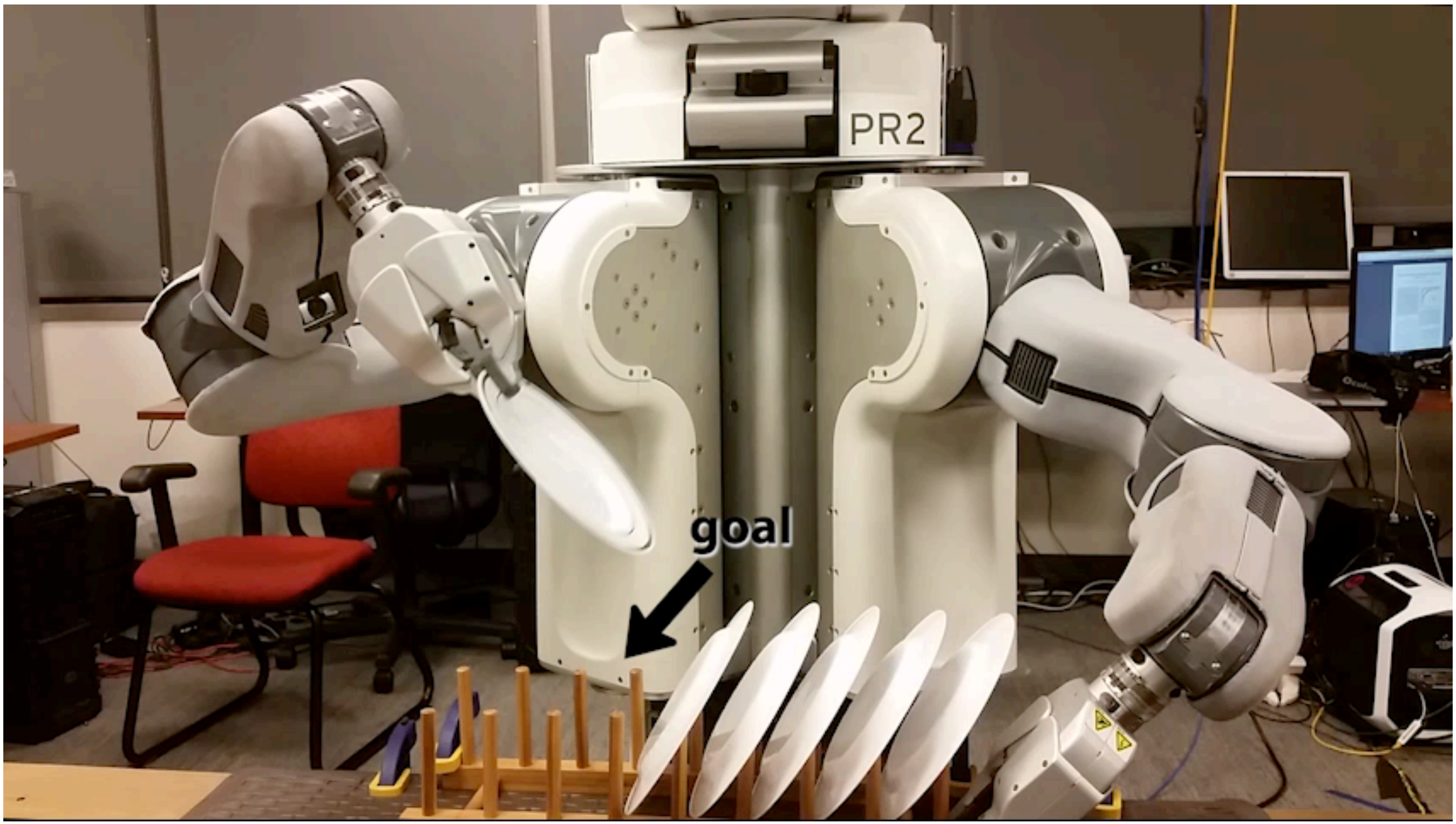
Pouring, GCL policy



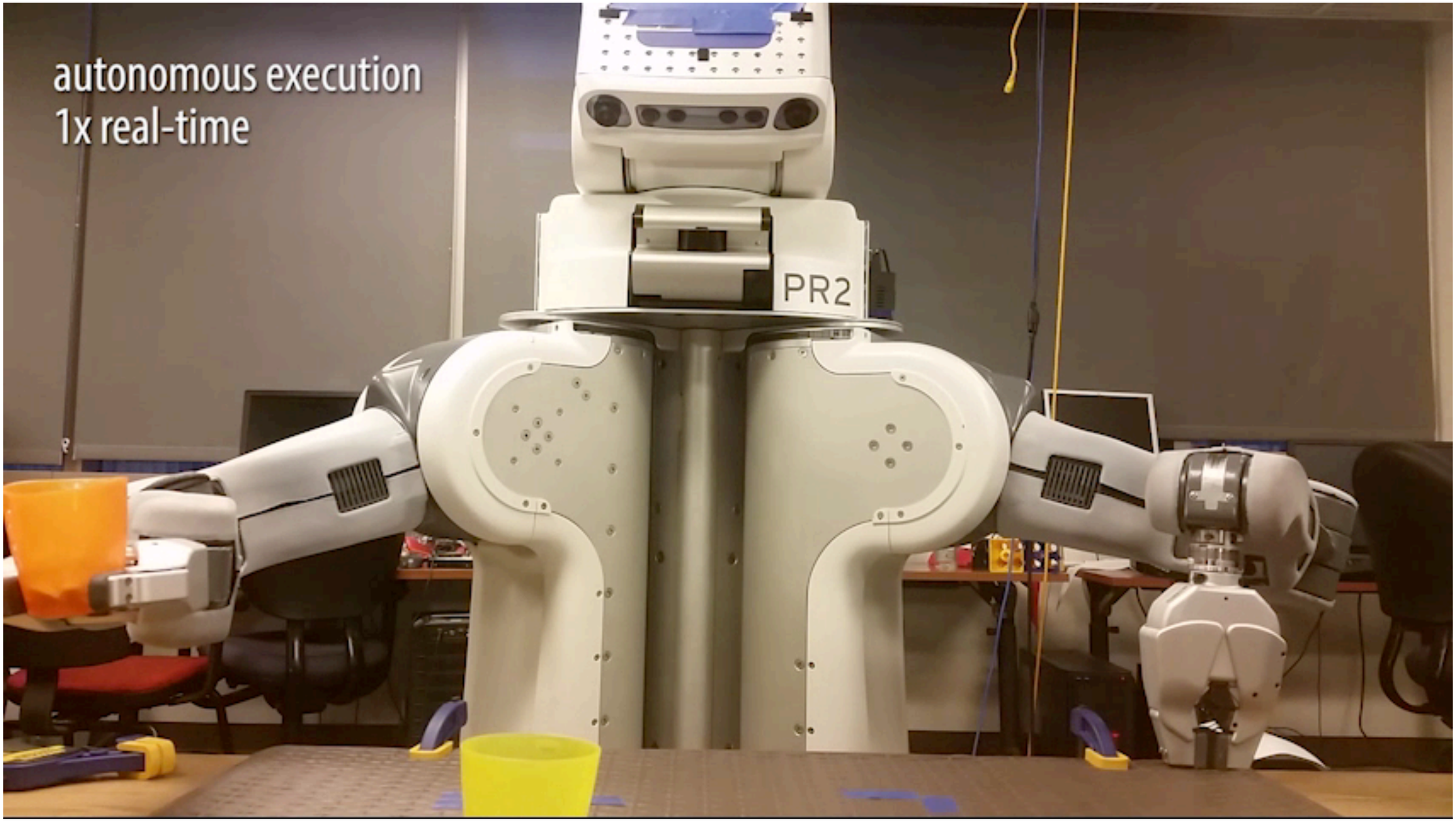
Conclusion: We can recover successful policies for new positions.

Is the reward function also useful for new scenarios?

Dish placement - GCL reopt.



Pouring - GCL reopt.



Note: normally the GAN discriminator is discarded

Guided Cost Learning & Generative Adversarial Imitation Learning

Strengths

- can handle unknown dynamics
- scales to neural net rewards
- efficient enough for real robots

Limitations

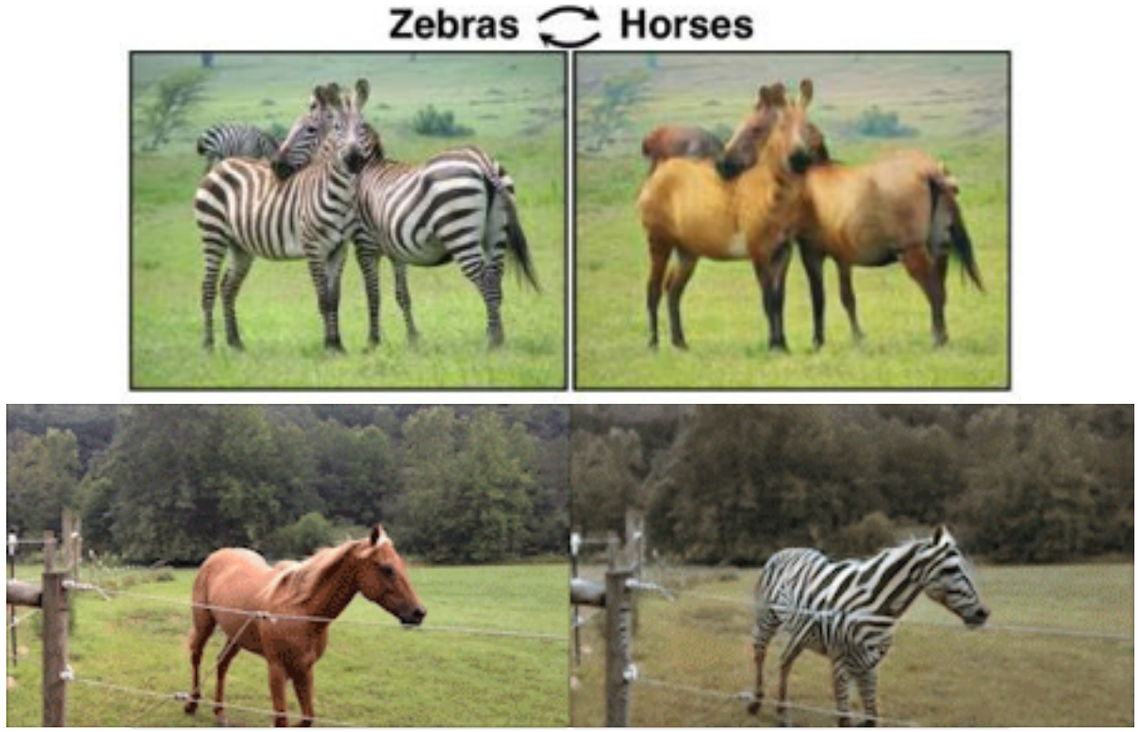
- adversarial optimization is hard
- can’t scale to raw pixel observations of demos
- demonstrations typically collected with kinesthetic teaching or teleoperation (first person)

Outline

1. A World without Rewards
2. A Probabilistic Model of Behavior
3. Application: Inverse RL
4. **GANs and Energy-Based Models**
5. Application: Soft-Q Learning

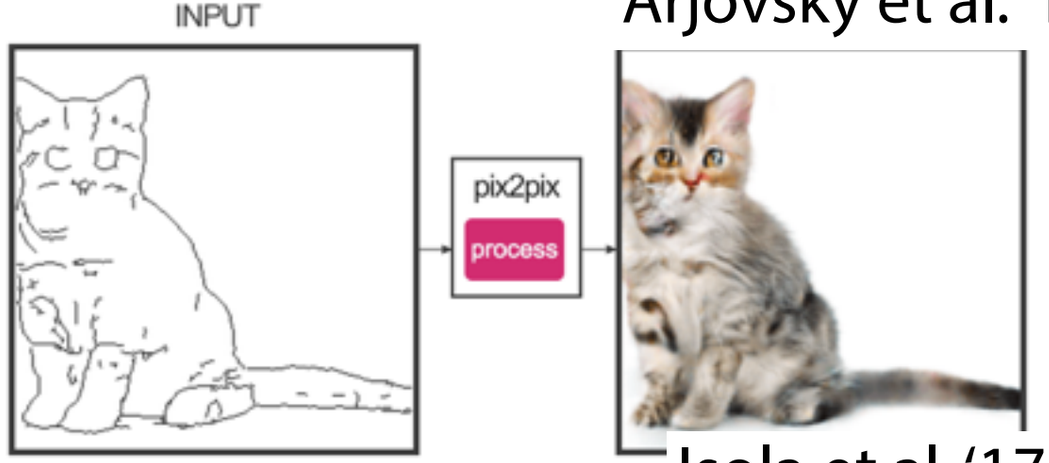
Generative Adversarial Networks

(Goodfellow et al. ’14)



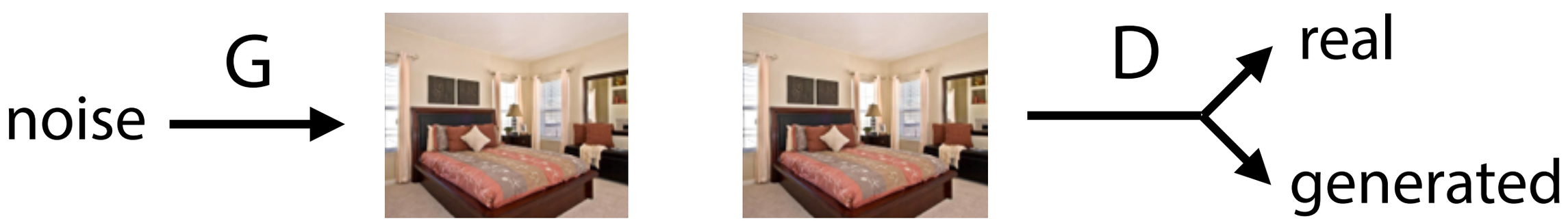
Zhu et al. ‘17

Arjovsky et al. ‘17



Isola et al. ‘17

Similarly, **GANs** learn an objective for generative modeling.



Connection between Inverse RL and GANs

trajectory τ $\longleftrightarrow$ **sample** x

policy $\pi \sim q(\tau)$ $\longleftrightarrow$ **generator** G

reward r $\longrightarrow$ **discriminator** D

discriminator

$$D^*(\tau) = \frac{p(\tau)}{p(\tau) + q(\tau)}$$

$$D_\theta(\tau) = \frac{\frac{1}{Z}\exp(r_\theta(\tau))}{\frac{1}{Z}\exp(r_\theta(\tau)) + q(\tau)}$$

discriminator only needs to learn data distribution,
θ independent of generator density

Finn*, Christiano*, Abbeel, Levine, arXiv '16

Connection between Inverse RL and GANs

trajectory τ ⟷ **sample x**

policy π~q(τ) ⟷ **generator** G

cost c ⟶ **discriminator** D

generator

$$\mathcal{L}_{\text{generator}}(q) = \mathbb{E}_{\tau\sim q}[\log(1 - D(\tau)) - \log(D(\tau))]$$

$$= \log Z - \mathbb{E}_{\tau\sim q}[r_\theta(\tau)] + \mathbb{E}_{\tau\sim q}[\log q(\tau)]$$

generator objective is **entropy-regularized RL**

Finn*, Christiano*, Abbeel, Levine, arXiv '16

GANs for training EBMs

MaxEnt IRL is an energy-based model

sampler q(x) $\longleftrightarrow$ **generator** G

energy E $\longrightarrow$ **discriminator** D

Use the **generator's density** q(x) to form a **consistent estimator** of the energy function

$$D_\theta(\mathbf{x}) = \frac{\frac{1}{Z}\exp(-E_\theta(\mathbf{x}))}{\frac{1}{Z}\exp(-E_\theta(\mathbf{x})) + q(\mathbf{x})}$$

Dai et al., ICLR submission '17

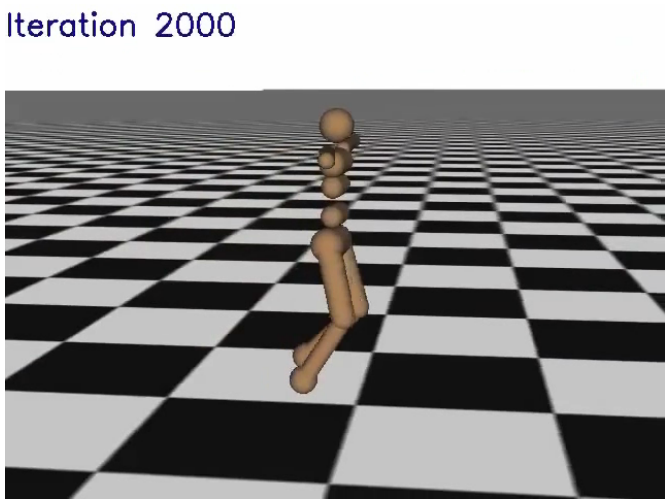
Kim & Bengio ICLR Workshop '16; **Zhao et al.** arXiv '16; **Zhai et al.** ICLR sub '17

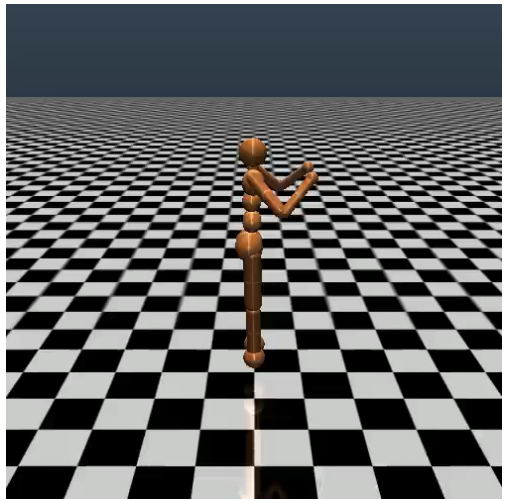
Finn*, Christiano*, Abbeel, Levine, arXiv '16

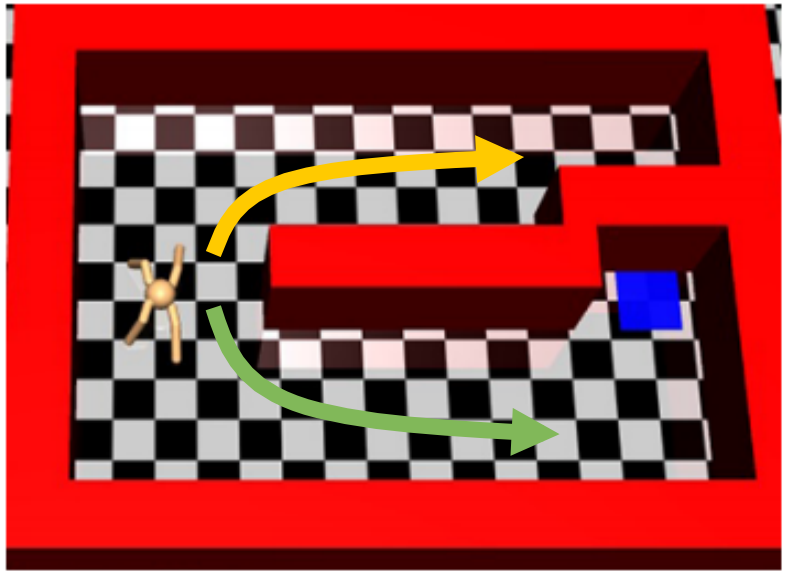
Outline

1. A World without Rewards
2. A Probabilistic Model of Behavior
3. Application: Inverse RL
4. GANs and Energy-Based Models
5. **Application: Soft-Q Learning**

Stochastic models for learning control



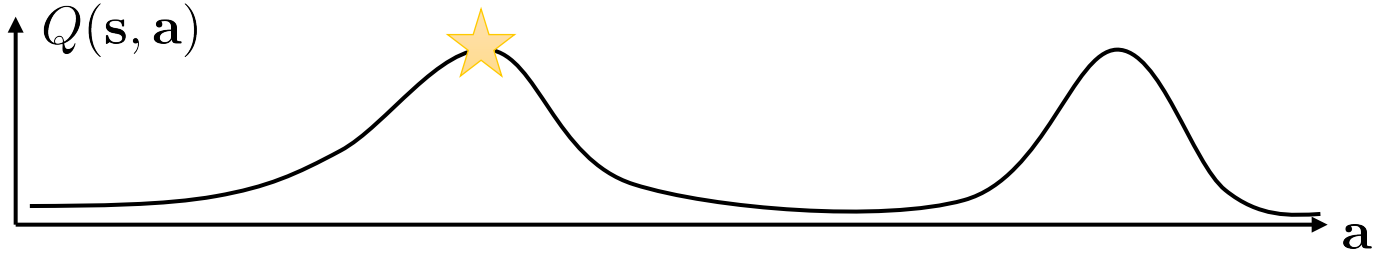


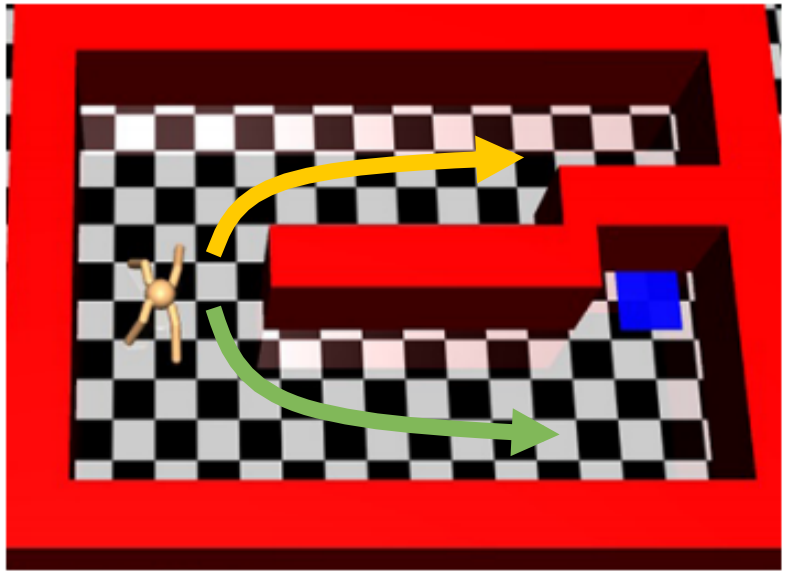


- How can we track *both* hypotheses?

Stochastic energy-based policies

Q-function: $Q(\mathbf{s}, \mathbf{a}) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$





$$\pi(\mathbf{a}|\mathbf{s}) \propto \exp(Q(\mathbf{s}, \mathbf{a}))$$

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t) - V_t(\mathbf{s}_t)) = \exp(A_t(\mathbf{s}_t, \mathbf{a}_t))$$

$$Q_t(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + E[V_{t+1}(\mathbf{s}_{t+1})]$$

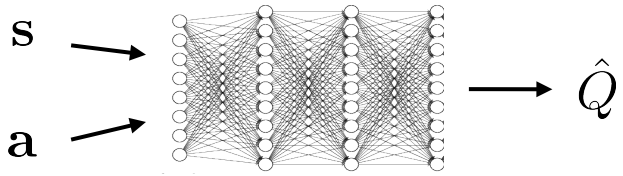
$$V_t(\mathbf{s}_t) = \log \int \exp(Q_t(\mathbf{s}_t, \mathbf{a}_t))\mathbf{a}_t$$

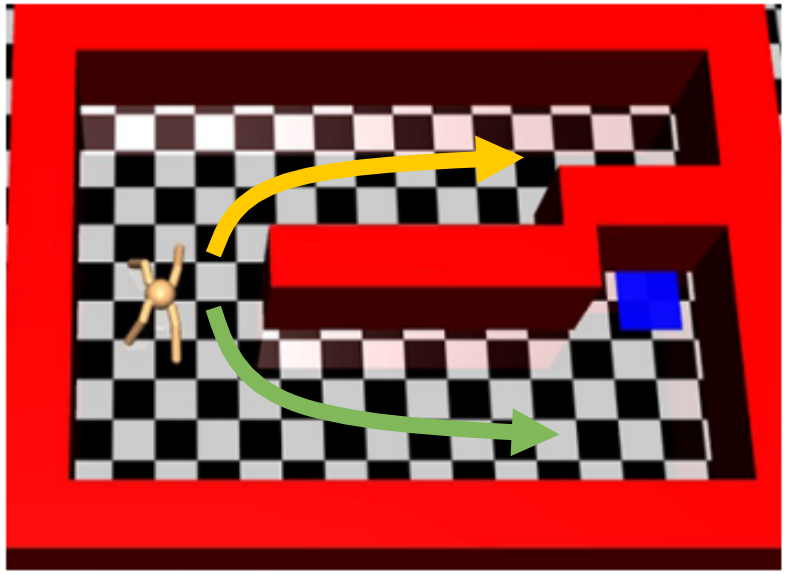
Tuomas Haarnoja

Haoran Tang

Soft Q-learning

Learned (neural network) Q-function: $Q_\theta(\mathbf{s}, \mathbf{a})$





Q-learning: $\theta \leftarrow \theta + \alpha \nabla_\theta Q_\theta(\mathbf{s}, \mathbf{a})(r(\mathbf{s}, \mathbf{a}) + \gamma V(\mathbf{s}') - Q_\theta(\mathbf{s}, \mathbf{a}))$

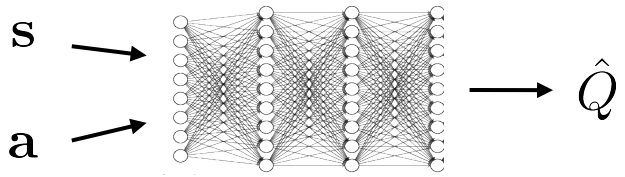
target value: $V(\mathbf{s}') = \max_{\mathbf{a}'} Q_\theta(\mathbf{s}', \mathbf{a}')$

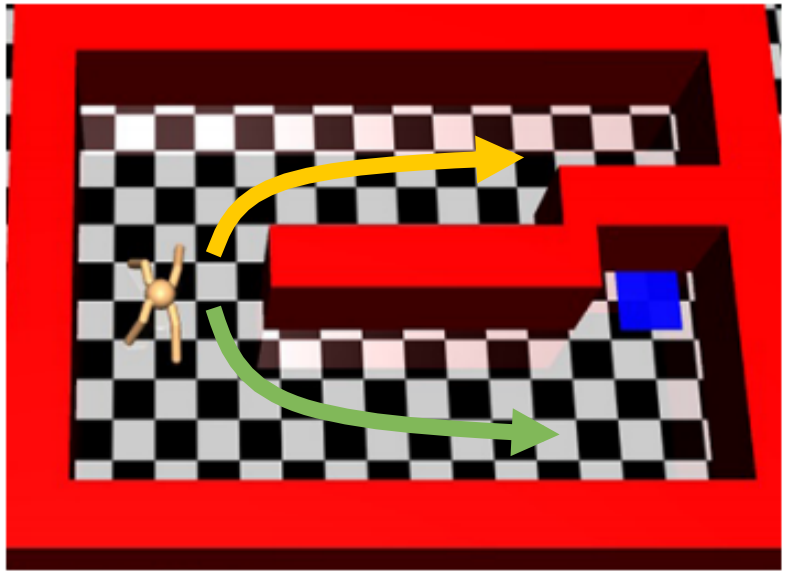
soft Q-learning: $\theta \leftarrow \theta + \alpha \nabla_\theta Q_\theta(\mathbf{s}, \mathbf{a})(r(\mathbf{s}, \mathbf{a}) + \gamma V(\mathbf{s}') - Q_\theta(\mathbf{s}, \mathbf{a}))$

target value: $V(\mathbf{s}') = \text{soft}\max_{\mathbf{a}'} Q_\theta(\mathbf{s}', \mathbf{a}') = \log \int \exp(Q_\theta(\mathbf{s}', \mathbf{a}'))d\mathbf{a}'$

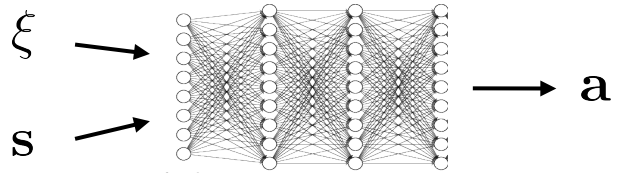
Tractable amortized inference for continuous actions

$\pi(\mathbf{a}|\mathbf{s}) \propto \exp(Q(\mathbf{s}, \mathbf{a}))$





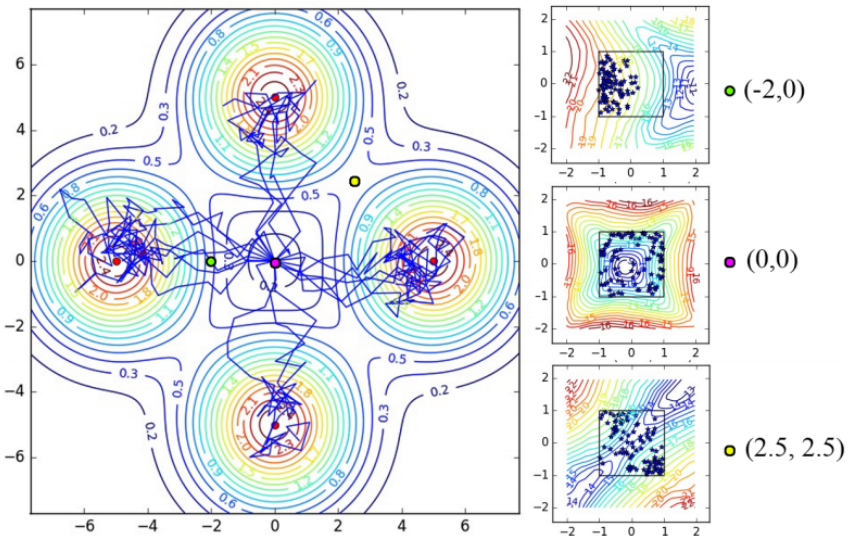
stochastic network:

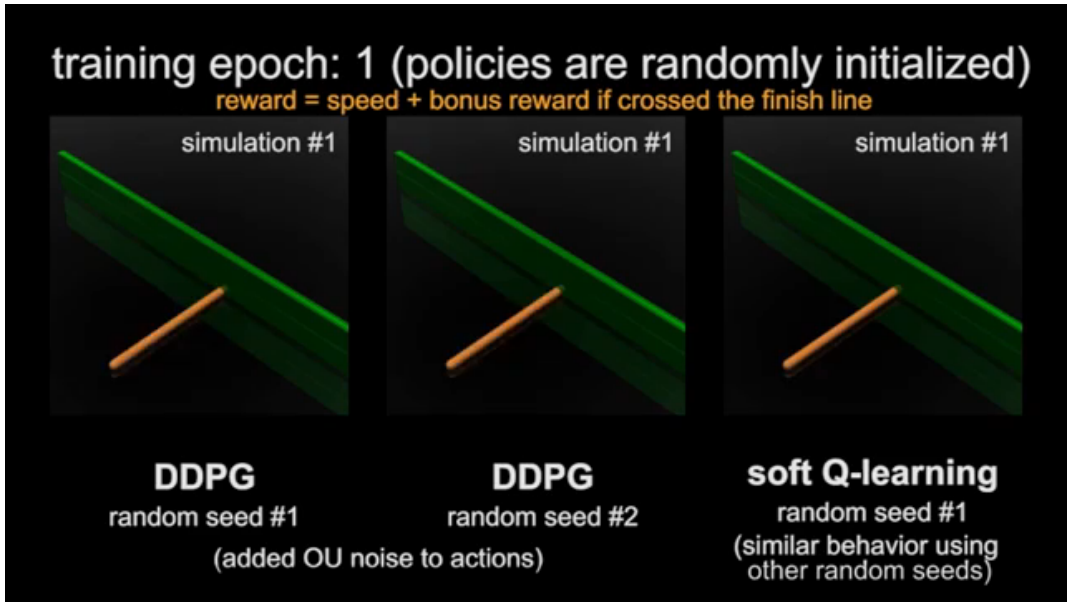


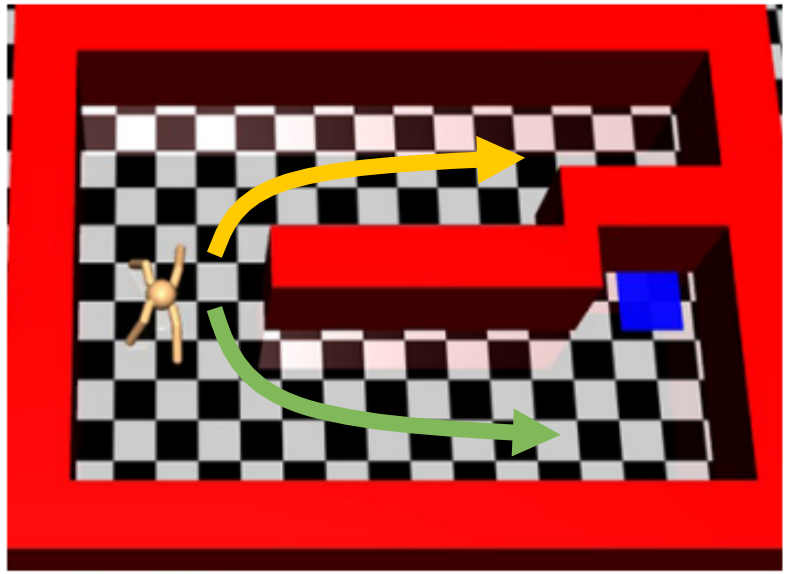
Trained with amortized SVGD to match $\pi(\mathbf{a}|\mathbf{s}) \propto \exp(Q(\mathbf{s}, \mathbf{a}))$

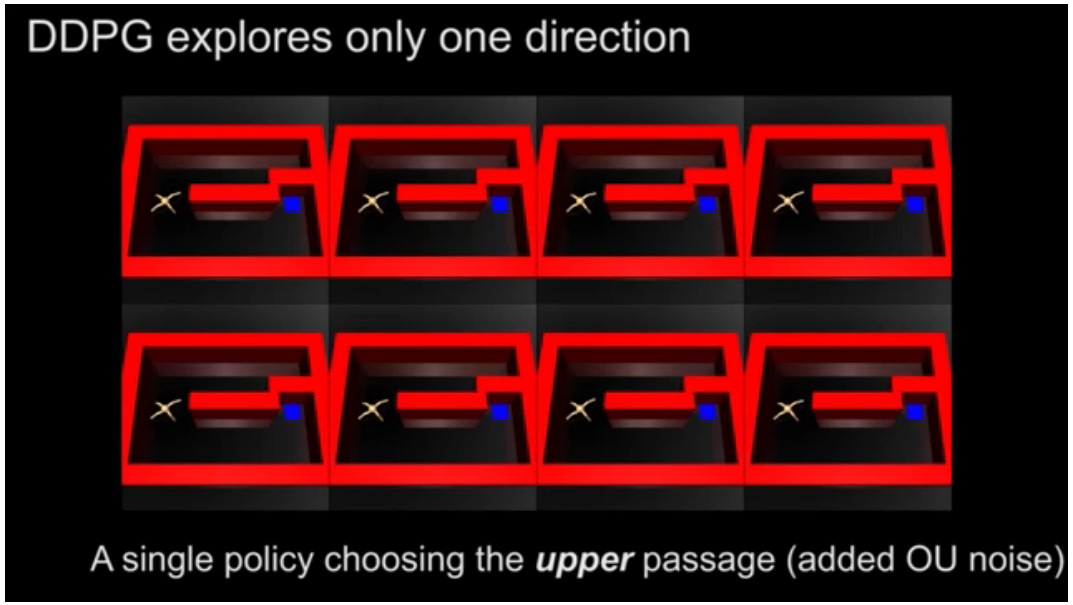
Wang & Liu, ‘17

Stochastic energy-based policies aid exploration

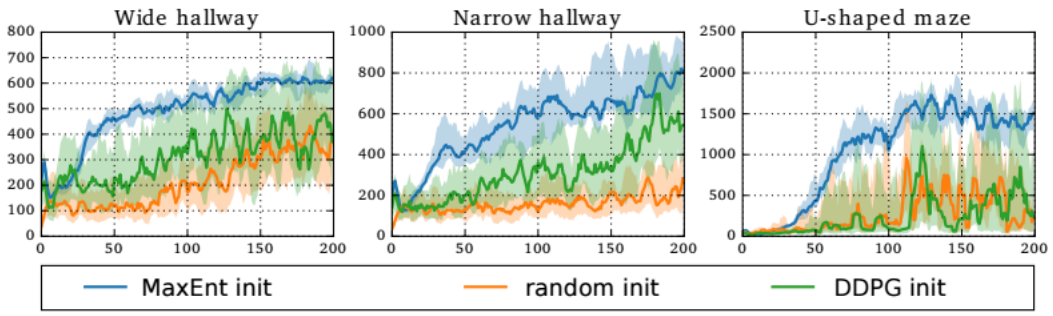




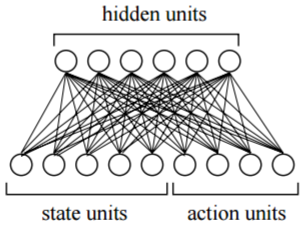


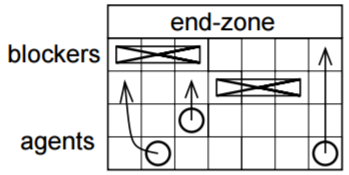


Stochastic energy-based policies provide pretraining



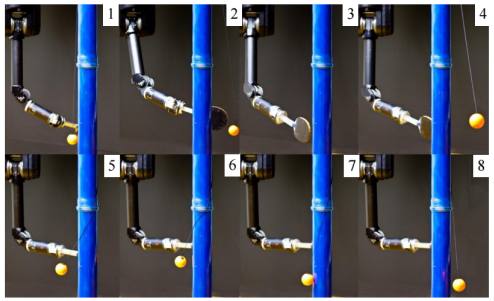
Stochastic Optimal Control & MaxEnt in RL

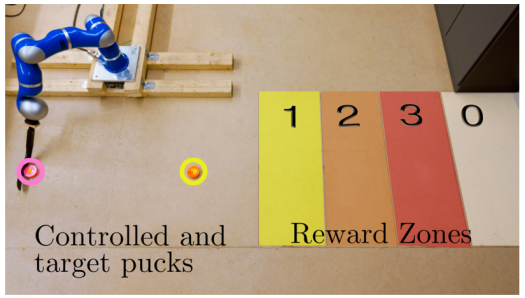




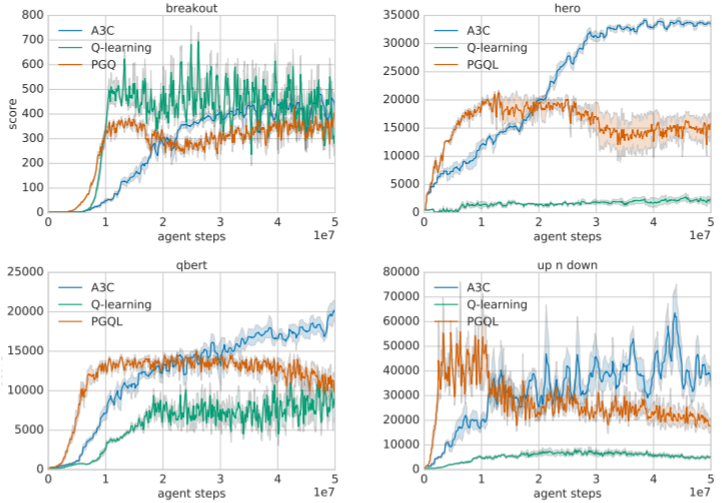
Sallans & Hinton. Using Free Energies to Represent Q-values in a Multiagent Reinforcement Learning Task. 2000.

Nachum et al. Bridging the Gap Between Value and Policy Based Reinforcement Learning. 2017.





Peters et al. Relative Entropy Policy Search. 2010.

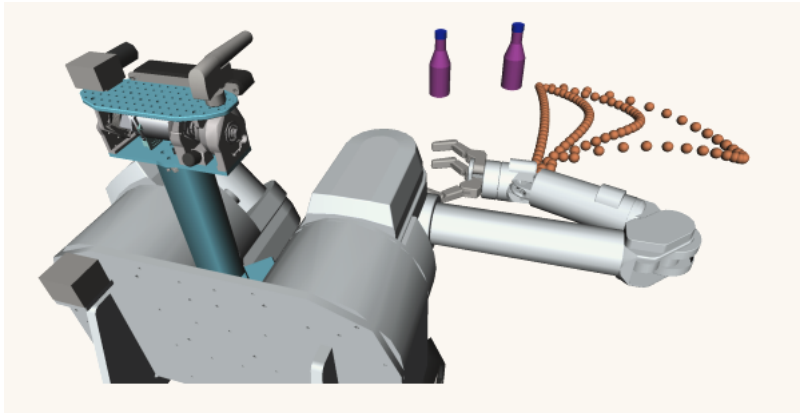


O'Donoghue et al. Combining Policy Gradient and Q-Learning. 2017

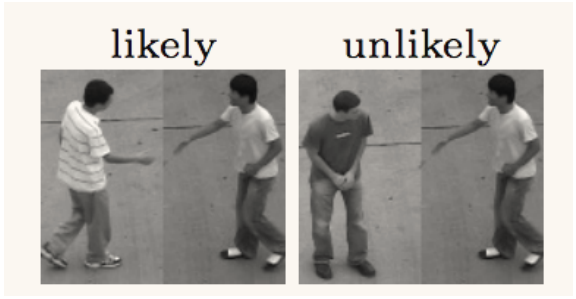
Applications of inverse reinforcement learning

(beyond robotic manipulation and control)

Ziebart et al. ‘08: Predict taxi driver route preferences



Dragan et al. ‘13: Generating human-legible motion



Kitani et al. ‘14: Model human pedestrian interactions

Input	what are you going to do ?
Vanilla-MLE	i ’m going to the movies .
Vanilla-Sample	get him outta here first !
REINFORCE	i ’m going to get you .
REGS Monte Carlo	i ’m going to get you a new car .

Input	tell me ... how long have you had this falling sickness ?
Vanilla-MLE	i ’m not a doctor .
Vanilla-Sample	well everything you did was totally untrue .
REINFORCE	i don ’t know how long it ’s been .
REGS Monte Carlo	A few months, I guess .

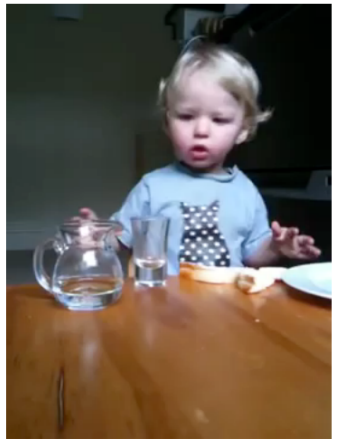
Li et al. ‘17: Learn objective for dialog generation

Concluding remarks

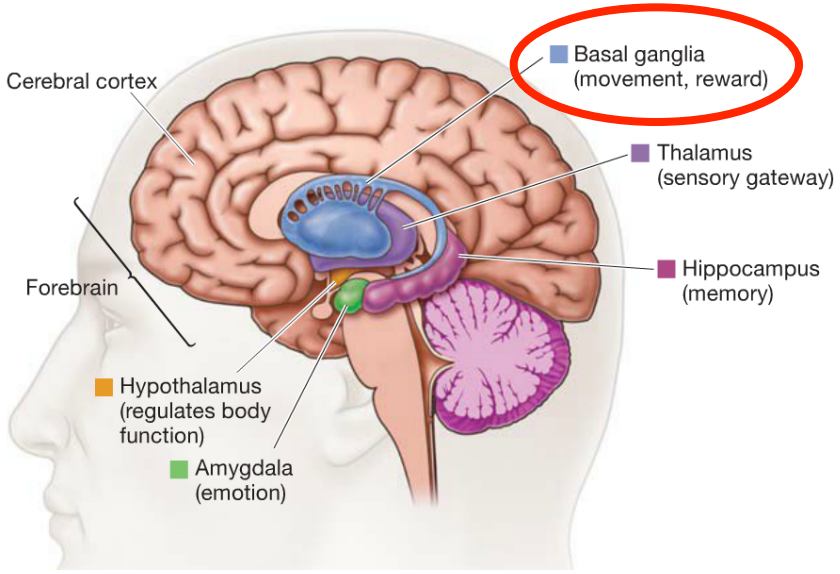
reward

Mnih et al. '15

reinforcement learning agent



what is the reward?



[–] **LazyOptimist** 32 points 5 days ago

> As human agents, we are accustomed to operating with rewards that are so sparse that we only experience them once or twice in a lifetime, if at all.